

## PLMSH\_18, Controlador Lógico Compacto Basado en el Microcontrolador MC9S08SH32

Antonio Salvá\*. Luis Altamirano\*\*. Víctor Sánchez \*\*\*.

\*Facultad de Ingeniería, Departamento de Control y Robótica.  
UNAM, D.F.04510, México, Teléfono: (52)-55-56223108  
salva@dctrl.fi-b.unam.mx

\*\*Facultad de Ingeniería, Departamento de Control y Robótica.  
UNAM, D.F.04510, México, Teléfono: (52)-55-56223108  
altamirano@dctrl.fi-b.unam.mx

\*\*\*Facultad de Ingeniería, Departamento de Control y Robótica.  
UNAM, D.F.04510, México, Teléfono: (52)-55-56223108  
victor@dctrl.fi-b.unam.mx

---

**Resumen**—En este trabajo se presenta un prototipo de controlador lógico denominado PLMSH\_18 (Programador Lógico Modular basado en el microcontrolador MC9S08SH32 de NXP/FREESCALE). El dispositivo desarrollado cuenta con 12 entradas binarias, 6 salidas binarias y puede manejar hasta 80 variables binarias internas. Para el desarrollo de aplicaciones de control lógico y secuencial (CLS), el dispositivo puede realizar diversos bloques funcionales, denominados módulos lógicos, que pueden ser entre otros: compuertas lógicas, temporizadores, contadores de eventos y secuenciadores de estados binarios. Para fines de la realización de un determinado CLS, el dispositivo cuenta con un lenguaje de programación textual denominado SIIL2, para el cual se desarrolló un compilador cruzado denominado COMSIIL2, el cual es parte de un software manejador denominado SWMANPLM, propio del PLMSH\_18 y otros dispositivos de su tipo diseñados previamente. El manejador SWMANPLM se ejecuta bajo WINDOWS en una computadora de propósito general, ligada vía puerto serie con el PLMSH\_18 y cuenta con, entre otras, las siguientes facilidades: compilación de programas fuente en SIIL2 con reporte de errores de sintaxis, carga y ejecución inmediata en el PLMSH\_18 de programas en SIIL2, que no contengan errores de sintaxis. El lenguaje textual SIIL2, el compilador cruzado COMSIIL2, y el software manejador SWMANPLM se diseñaron en el Departamento de Control y Robótica de la Facultad de Ingeniería de la UNAM campus pedregal, y para ello se emplearon herramientas de software y hardware para aprendizaje y desarrollo de aplicaciones basadas en microcontroladores, diseñadas previamente en esa dependencia.

**Palabras clave:** Automatización, Control Lógico, Control Secuencial, PLC, Microcontrolador, Módulo Lógico, PCB, MCU.

---

### 1. INTRODUCCIÓN

El PLMSH\_18 es un controlador lógico compacto basado en el microcontrolador (MCU) MC9S08SH32 de FREESCALE, que cuenta con doce entradas binarias, seis salidas binarias y puede manejar hasta 80 variables binarias internas. El dispositivo está construido empleando para ello una placa de circuito impreso (PCB) de doble cara de 12 x 12 cm y está integrado por los siguientes cinco bloques funcionales:

1. Computadora central (CC) realizada con la tarjeta FACIL\_08SH basada en el MCU MC9S08SH32.
2. Hardware de optoacoplamiento de entrada (HOE) entre borneras asociadas con cada una de las doce entradas binarias del dispositivo y bits de puertos de entrada del MCU asociados con éstas.

3. Hardware de interfaz de salida (HIS) entre bits de puertos de salida del MCU y relevadores que validan cada una de las seis salidas binarias del dispositivo.
4. Regulador conmutado (RC) para la obtención de la tensión de polarización del MCU (5 V), a partir de una fuente externa de 24 VDC.
5. Módulo de enlace serie (MES) entre el dispositivo y la computadora anfitriona donde se ejecuta el software manejador SWMANPLM.

Las entradas están presentadas en dos grupos denominados como grupo 0 y grupo 1, el primero abarca 8 entradas y el segundo abarca 4 entradas; de esta forma cada una de las doce entradas se denota como E<sub>x</sub>y, donde “y” define el número de bit asociado y “x” podrá ser 0 ó 1.

Las salidas están agrupadas en un conjunto de 6 de éstas y se denotan como S00 a S05.

Por otra parte, las variables binarias intermediarias se denotan como  $I_{xy}$ , donde “y” podrá estar comprendidas entre 0 y 7, y “x” estará entre 0 y 9.

En la figura 1 se muestra un diagrama de bloques simplificado del PLMSH\_18.

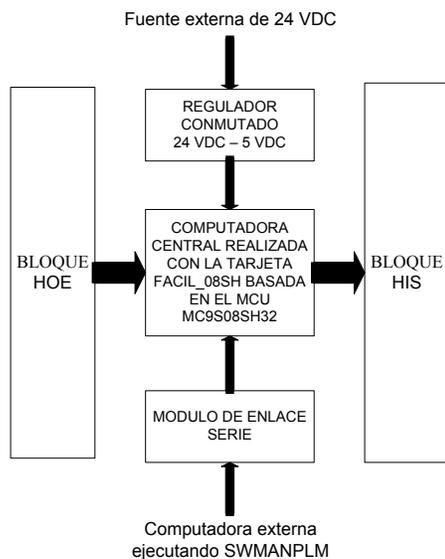


Fig. 1. Diagrama de bloques simplificado del PLMSH\_18.

El PLMSH\_18 es una evolución de dispositivos de su tipo previamente diseñados en el Departamento de Control y Robótica de la Facultad de Ingeniería de la UNAM. Un esbozo general acerca de estos prototipos anteriores puede verse en (Salvá, 2015c).

## 2. ASPECTOS DEL HARDWARE DEL PLMSH\_18

En esta sección se describen aspectos básicos acerca de los componentes de hardware que integran al PLMSH\_18, describiendo, en lo general, como se realizaron los bloques funcionales que lo integran, así como también la interacción de éstos con el MCU que realiza la CC.

### 2.1 CC del PLMSH\_18

Para realizar la computadora central del dispositivo se empleó la tarjeta FACIL\_08SH, basada en el MCU MC9S08SH32, ésta es una evolución compacta de otra tarjeta anterior con el mismo MCU y software de base para desarrollo, denominada MINICON\_08SH. Ambas tarjetas fueron previamente diseñadas por uno de los autores, para ser usadas en el aprendizaje y desarrollo de aplicaciones basadas en dicho MCU. Para ver aspectos acerca del software de base y entorno para desarrollo propios de ambas tarjetas, véase (Salvá 2016). De hecho, el PLMSH\_18 es una aplicación de la tarjeta FACIL\_08SH. El hardware del PLMSH\_18 está integrado por dos componentes, uno es una placa madre

(PM), que contiene cuatro de los cinco bloques que integran al dispositivo, y el otro es la tarjeta FACIL\_08SH, que es desmontable de la PM. En la figura 2 se muestra la tarjeta FACIL\_08SH, en la figura 8 se aprecia ésta como componente funcional del PLMSH\_18.



Fig. 2. Tarjeta FACIL\_08SH empleada para realizar la CC del PLMSH\_08.

### 2.2 Hardware de optoacoplamiento para las entradas del PLMSH\_18

Como es sabido, en general el tipo de señales binarias de entrada a un controlador lógico, puede verificar cada uno de los dos niveles lógicos posibles, con la presencia o ausencia de una tensión eléctrica de CA o CD, véanse (Erickson, 2011) y (Bolton, 2015). Para el PLMSH\_18 se estableció que se empleara una tensión de CD para estos fines. Así, la existencia de una tensión de 24 Volts de CD en las entradas es interpretada como uno lógico, y la ausencia de tensión en las entradas es interpretada como cero lógico, como es el caso en los módulos de entradas de los controladores lógicos comerciales, que empleen niveles de CD para los fines aquí descritos.

Con el fin de aislar eléctricamente el entorno de lógica de 24 Volts propio de las entradas físicas del dispositivo, del entorno de lógica TTL propia del MCU que realiza la CC, cada una de las entradas del dispositivo cuenta con un optoacoplador colocado entre la entrada física y el bit de entrada del MCU, asociado con la entrada física en cuestión. Además, se diseñó circuitería para que cada una de las entradas individualmente pueda ser configurada de modo que ésta sea de tipo sumidero o fuente. Esto es relevante cuando se usan sensores de tres hilos que pueden ser, acorde con su transistor de salida de tipo NPN, o bien de tipo PNP. Para un sensor tipo NPN se requerirá que la entrada asociada sea de tipo fuente, por otro lado si el sensor es de tipo PNP la entrada asociada deberá ser de tipo sumidero.

Para fines de la preconfiguración de una determinada entrada, para que ésta sea de tipo fuente o sumidero, existen, para cada una de ellas, postes donde han de colocarse puentes en una determinada posición según sea el caso. En la figura 3 se muestra una entrada preconfigurada para que ésta sea de tipo fuente con un sensor de tipo NPN conectado. En la figura 4 se muestra lo propio para una entrada preconfigurada para que sea de tipo sumidero, mostrándose ahí la conexión de un sensor de tipo PNP.

En las figuras 3 y 4 “Exy” denota genéricamente a cualquiera de las doce entradas del PLMSH\_18. Ptx denota genéricamente a cada uno de los bits de entrada del MCU, asociados con cada una de las entradas físicas. Éstos no se detallan aquí por razones de espacio.

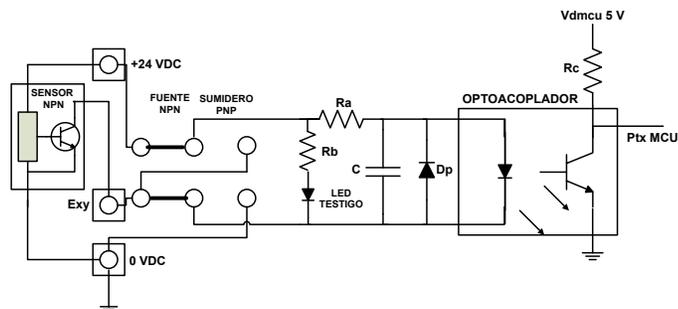


Fig. 3. Hardware de entrada preconfigurada como de tipo fuente.

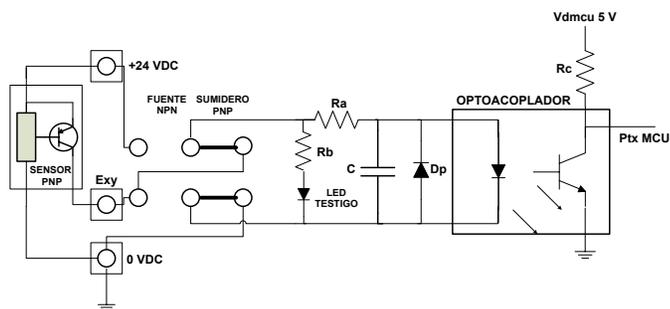


Fig. 4. Hardware de entrada preconfigurada como de tipo sumidero.

### 2.3 Hardware de interfaz para las salidas del PLMSH\_18

Cada una de las 6 salidas del PLMSH\_18 tiene asociado un relevador que puede manejar un máximo de 5 amperes para sus contactos. Cuando una determinada salida es uno lógico, los contactos del relevador asociado estarán cerrados, existiendo continuidad eléctrica entre un borne común y el borne asociado con la propia salida. Para las salidas S00, S01 y S02 el borne común se denomina COMSAL1 y para las salidas S03, S04 y S05, éste se denomina como COMSAL2. Para accionar los relevadores de salida desde el MCU se empleó el circuito integrado (CI) ULN2003A de Texas Instruments, el cual contiene 7 arreglos Darlington, con sendos trios de resistencias de polarización y un diodo para protección de los transistores, dado que éstos conmutan cargas inductivas, que es el caso de las bobinas de los relevadores de salida. En la figura 5 se muestra en forma genérica el circuito asociado con las salidas del PLMSH\_18. El arreglo Darlington, los componentes RSn1, RSn2, RSn3 y el diodo DSn están dentro del CI mencionado. PTyn denota genéricamente a los bits de salida del MCU que están asociados con las salidas físicas del dispositivo. No se detalla aquí explícitamente cuales son, por razones de espacio. Por otra parte, los dos fusibles empleados son de 10 amperes.

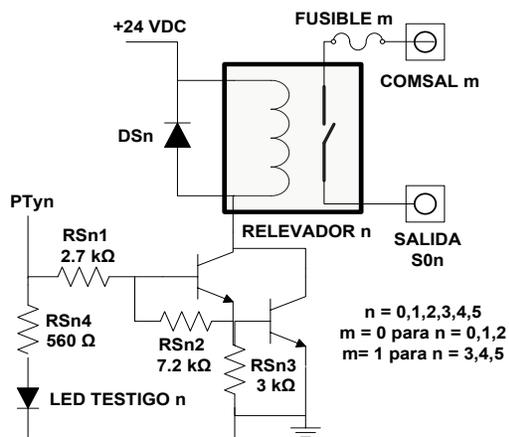


Fig. 5 Circuito genérico para las salidas del PLMSH\_18.

### 2.4 Módulo de enlace serie

Para fines del enlace con la computadora anfitriona donde corre el software manejador SWMANPLM, se empleó el bien conocido circuito que se muestra en la figura 6, que emplea el CI MAX 3232 para efectuar los cambios de nivel TTL – RS232 y RS232 – TTL requeridos. Cabe señalar además que el dispositivo cuenta con postes, no mostrados aquí, para conexión de las señales de transmisión y recepción propias de un adaptador USB-SERIE/TTL, que se emplearía para interactuar con el PLMSH\_18 desde una computadora que no cuente con puerto serie RS232.

### 2.5 Fuente de poder del PLMSH\_18

El PLMSH\_18 emplea dos voltajes de CD para su operación (24 y 5 VDC), el primero se obtiene a partir de una fuente externa, el segundo se genera empleando al CI LM2576-ADJ de Texas Instruments, que es un regulador conmutado que admite como entrada tensiones entre 7 y 40 VDC y presenta en su salida una tensión de 5 VDC, para esto, además del propio CI, se requieren otros componentes.

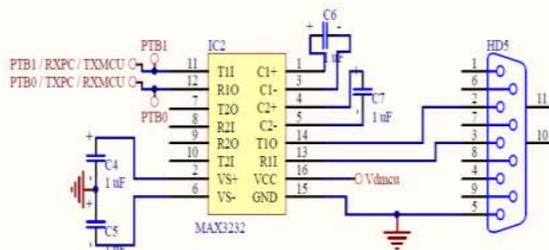


Fig. 6 Hardware del módulo de enlace serie (MES).

En la figura 7 se muestra el circuito empleado, donde se aprecia que la entrada es una tensión de 24 VDC, provista por la fuente externa, denominada como Va y que en la salida se

tiene una tensión de + 5 VDC denominada  $V_{dmcu}$ , que es la empleada para polarizar la CC y el lado del MCU de los optoacopladores.

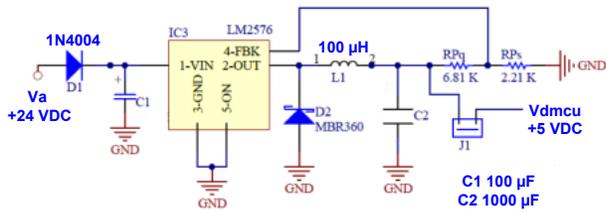


Fig. 7. Fuente de poder del PLMSH\_18.

### 2.6 Aspecto del primer prototipo del PLMSH\_18

En la figura 8 se muestra una fotografía del primer prototipo del PLMSH\_18. Se aprecian ahí los siguientes componentes del dispositivo:

1. Computadora Central
2. Optoacopladores de entrada
3. Relevadores de salida
4. Regulador conmutado
5. Punto de conexión de enlace serie
6. Regleta con bornes de entrada y fuente externa
7. Puentes de configuración fuente/sumidero para las entradas
8. Regleta con bornes asociados con las salidas

### 3. ENTORNO PARA DESARROLLO CON EL PLMSH\_18

Para desarrollar aplicaciones de control lógico, el PLMSH\_18 cuenta con el lenguaje textual de programación SIIL2, procesable con el compilador cruzado COMSIL2, el cual es parte del software manejador SWMANPLM, tanto el compilador como el software manejador fueron diseñados previamente a este trabajo, véase (Salvá, 2015b). Así, el entorno para desarrollo está conformado por una computadora de propósito general donde se ejecuta el software manejador, enlazada vía serie con el dispositivo, véase la figura 9.

#### 3.1 Módulos lógicos

Para fines de la validación del sistema lógico asociado con un determinado Control Lógico y Secuencial (CLS), el PLMSH\_18 puede realizar los siguientes módulos lógicos (ML):

- Seguidores lógicos
- Inversores lógicos
- Un total de 168 variaciones de compuertas lógicas AND, NAND, OR, NOR, XOR y XOR negada, de 2, 3 y 4 entradas

- Un total de 8 variaciones de temporizadores monodisparo con disparo por nivel (one-shot)
- Un total de 8 variaciones de temporizadores monodisparo con disparo por flanco (one-shot)
- Temporizadores con retardo a la activación (TRA, on-delay)
- Temporizadores con retardo a la desactivación (TRD, off-delay)
- Un total de 2 variaciones de temporizadores astables
- Un total de 8 variaciones de contadores de eventos
- Un total de 12 variaciones de comparadores de cuenta de contador de eventos
- Un total de 8 variaciones de Flip-Flops RS asíncronos
- Un total de 16 variaciones de secuenciadores de estados de un bit (drummers)

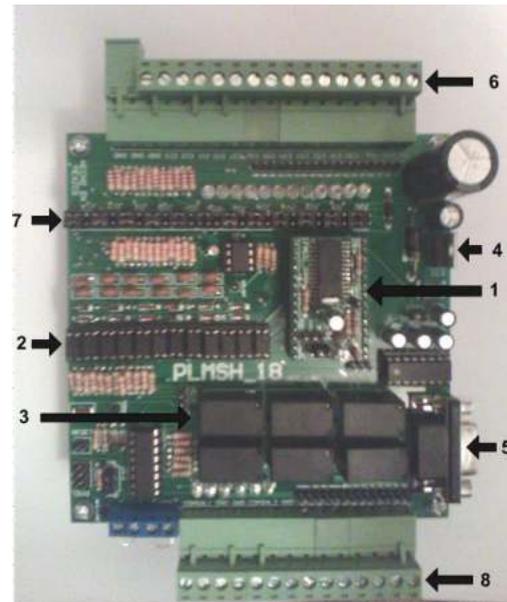


Fig. 8. Primer prototipo del PLMSH\_18.

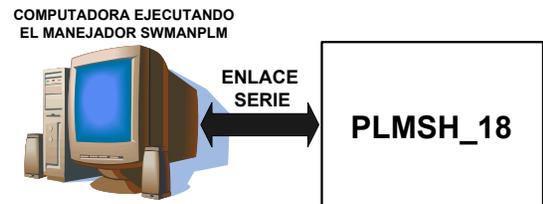


Fig. 9. Entorno para desarrollo con el PLMSH\_18.

En la realización de un determinado CLS intervendrán uno o varios ML interconectados de alguna forma.

Éstos se realizan virtualmente mediante código ejecutable en la CC del PLMSH\_18. Dicho código se genera a partir de un programa fuente escrito en SIIL2, empleando para ello al

compilador COMSIL2, que es parte del SWMANPLM. Detalles generales acerca de cómo se desarrolló el compilador pueden verse en (Salvá, 2015b).

### 3.2 Programación del PLMSH\_18 empleando SIIL2

Para fines del desarrollo y depuración del programa en SILL2 asociado con una determinada aplicación, el usuario final debe escribir en el editor del SWMANPLM el programa fuente asociado con ésta. Después de esto, debe compilarlo y si no hubo errores sintácticos, lo puede cargar y ejecutar en el PLMSH\_18. Un programa en SIIL2 está integrado por dos subprogramas uno denominado subprograma principal (SPP), que se ejecuta cíclicamente; y otro denominado subprograma temporizado (SPT), que se ejecuta cada 10 ms mediante una interrupción de overflow del MCU que realiza la CC del PLMSH\_18. El armado del código ejecutable con sus componentes en el SPP y SPT es hecho por el compilador COMSIL2; el usuario final simplemente debe escribir el programa fuente siguiendo el siguiente formato:

```
Inprog 'Sentencia delimitadora de inicio del subprograma principal
Sentencias declaratorias de los ML implicados en la aplicación
Finpp 'Sentencia delimitadora del final del subprograma principal
'Sentencias delimitadoras de código a ejecutarse con 'una temporalidad base de 10 ms.
Inmodi 'Inicio del subprograma temporizado
Finmodi 'Fin del subprograma temporizado
```

Nótese el uso del caracter apostrofe para delimitar comentarios. Las sentencias inmodi y finmodi deberán ir siempre colocadas en la forma mostrada. Una vez que el programa asociado con una determinada aplicación ha sido validado, para que éste se ejecute de manera autónoma sin el concurso del SWMANPLM, bastará pasar un interruptor de un polo dos tiros presente en la CC (tarjeta FACIL\_08SH), de la posición *Mon* a la posición *Ejaut*.

### 3.3 Formato de las sentencias declaratorias de los ML

La forma genérica para la declaración en SIIL2 de los ML que puede realizar el PLMSH\_18 es:

CODM#N E1,.. En,.. S1,.. Sm, D1,.., Dq, CADBI

Donde, CODM es una cadena de caracteres que simboliza la función efectuada por el módulo, por ejemplo, para una compuerta 'or' de tres entradas CODM será 'OR3' y para un temporizador monodisparo con disparo por nivel será 'TEMPOC'; N, es el número asociado con el módulo, ya que todos los ML de un mismo tipo de una aplicación, deben ser numerados. Los demás argumentos podrán representar

diversos parámetros, entre otros: definición de las entradas y salidas asociadas con el ML, tiempos asociados con un temporizador y nivel de verificación de entradas o salidas del ML en cuestión.

Para detalles acerca de la sintaxis propia de la declaración de cada uno de los ML realizables con el PLMSH\_18, puede verse (Salvá, 2015a).

## 4. EJEMPLO DE APLICACIÓN

Para fines ilustrativos, en esta sección se muestra un ejemplo sencillo de uso del PLMSH\_18, que consiste en el control lógico de un sistema Cisterna – Tanque Alto con histéresis en ambos depósitos. Esto para un líquido conductor. Para la realización de este CLS se emplean cuatro entradas (E00, E01, E02 y E03) y una salida (S00) del dispositivo. Los sensores empleados son de tipo pnp, y presentarán la tensión de 24 VDC cuando el nivel del líquido rebase o sea igual a la posición del electrodo asociado. Por lo tanto, las cuatro entradas se deben configurar en modo sumidero, véase la sección 2.2. En la figura 10 se muestra un esquema del CLS de este ejemplo, apreciándose ahí los conexiones de potencia y los propios de los sensores y el actuador al PLMSH\_18.

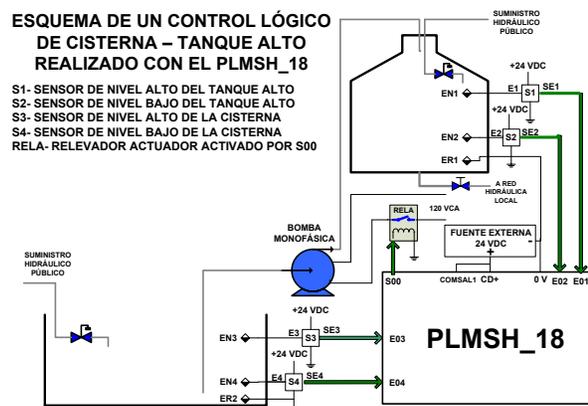


Fig. 10. Esquema del CLS asociado con el ejemplo ilustrativo.

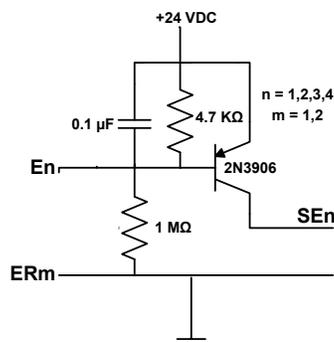


Fig. 11. Esquema genérico de los sensores de nivel.

Además, se requiere que una vez que el nivel del líquido haya rebasado el nivel del sensor asociado con el electrodo EN1, deberán transcurrir 30 segundos antes de que la bomba se desconecte. En la figura 11 se muestra el esquema genérico asociado con los cuatro sensores de nivel empleados. El sistema lógico asociado con el CLS de este ejemplo se muestra en la figura 12. Ahí se ve que éste se realiza mediante el concurso de cuatro módulos que son: dos latches (flip-flops asíncronos en la nomenclatura de SIIL2), un temporizador con retardo a la desactivación y una compuerta 'and' de dos entradas. Se aprecia el uso de las variables internas I00, I01 e I02. En la figura se muestra, para fines ilustrativos debajo de cada bloque funcional, la declaración en SIIL2 asociada. Por razones de espacio, no se explica aquí la sintaxis de estas declaraciones, esto puede verse con amplitud en (Salvá, 2015a).

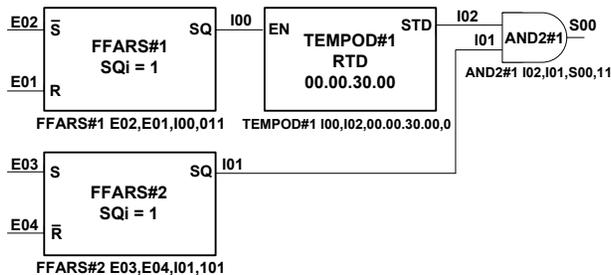


Fig. 12. Diagrama de bloques del CLS del ejemplo de aplicación.

A continuación se muestra el programa en SIIL2 asociado con este ejemplo.

```
Inprog
ffars#1 e02, e01,i00,011
ffars#2 e03, e04,i01,101
tempod#1 i00, i02, 00.00.30.00,0
and2#1 i02,i01,s00,11
finpp
```

```
inmodi
finmodi
```

## 6. CONCLUSIONES

Se presentó en este trabajo el desarrollo y funcionalidad del controlador lógico compacto PLMSH\_18, que es una evolución de otros dispositivos anteriores de su tipo diseñados previamente. Es importante destacar que la meta objetivo de esta aventura técnica, no fue emular *a pie juntillas* la funcionalidad tanto del software como del hardware de un determinado PLC comercial, sino el llegar a contar con un dispositivo compacto para el desarrollo de aplicaciones de control lógico, que tuviera la capacidad de realizar, en lo fundamental, las funciones lógicas para control lógico de uso más frecuente en la industria, y que además, esto se hiciera bajo el lenguaje propietario SIIL2. Para estos fines se llevaron a cabo, entre otras, las siguientes acciones:

Se adecuó el compilador COMSILL2 acorde con el perfil del MCU MC9S08SH32, empleado para realizar la CC del dispositivo; se diseñó circuitería de modo que cada una de las entradas del dispositivo pudiera individualmente ser configurada como de tipo sumidero, o bien como de tipo fuente, esta característica del dispositivo, da flexibilidad para fines de la selección de los sensores requeridos por una determinada aplicación; se diseñó la tarjeta FACIL\_08SH, de modo que ésta, además de ser parte integrante desmontable del PLMSH\_18, pudiera ser empleada individualmente para aprendizaje y desarrollo de aplicaciones con el MCU MC9S08SH32.

Trabajos futuros potenciales pueden ser, entre otros, los siguientes: Desarrollar un ambiente gráfico para que el dispositivo pueda programarse bajo diagramas de escalera; realizar la CC del dispositivo con un MCU basado en el procesador ARM CORTEX M4. A la fecha se cuenta ya con un ensamblador propietario para ese procesador, que sería la piedra angular para la adecuación del compilador COMSILL2 a este último procesador mencionado.

## 6. REFERENCIAS

Bolton, W. *PROGRAMMABLE LOGIC CONTROLLERS* (2015). Newnes. EUA.

Erickson, K. *Programmable Logic Controllers: An Emphasis on Design and Application. Second Edition* (2011). Dogwood Valley PressNewnes. EUA

Salvá, A. (2015a). GUÍA BÁSICA DE SIIL2. Descargar archivo: <http://dctrl.fi-b.unam.mx/~salva/gbsiil2.pdf>

Salvá, A, L. Altamirano (2015b). SIIL2, Lenguaje de Programación Textual para Control Lógico. MEMORIA DE CNCA 2015, realizado en Cuernavaca Morelos, México en octubre de 2015.

Salvá, A, L. Altamirano (2015c), L Álvarez, A Herrera. PLM3, Controlador Lógico Programable para Auxilio Didáctico. MEMORIA DE CNCA 2015, realizado en Cuernavaca Morelos, México en octubre de 2015.

Salvá, A (2016). AIDA08SH AMBIENTE INTEGRADO PARA DESARROLLO Y APRENDIZAJE CON MICROCONTROLADORES MC9S08SH32 HABILITADOS COMO DISPOSITIVOS CHIPBAS8SH. Descargar archivo: [http://dctrl.fi-b.unam.mx/~salva/guia\\_aida08sh\\_cn2.pdf](http://dctrl.fi-b.unam.mx/~salva/guia_aida08sh_cn2.pdf)